IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	§	Group Art Unit: 2163
	§	
John Colgrove, et al.	§	Examiner: Brown, Sheree N.
	§	
	§	Atty. Dkt. No.: 5760-14900
	§	VRTS0526
Serial No. 10/750,597	§	
	§	
	§	
Filed: December 31, 2003	§	
	§	
For: MULTI-CLASS STORAGE MECHANISM	§	
	§	
	§	
	§	
	§	
	§	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Panel Decision mailed October 11, 2006, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal. **No extension of time should be required since this Appeal Brief is filed within one month of the Notice of Panel Decision.**

I. REAL PARTY IN INTEREST

The subject application is owned by VERITAS Operating Corporation, which is a wholly owned subsidiary of Symantec Corporation.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-41 stand finally rejected. The rejection of claims 1-41 is being appealed. A copy of claims 1-41 as currently pending is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a system including a processor and memory. The memory comprises program instructions executable by the processor to implement file system software including a multi-class storage mechanism. The multi-class storage mechanism is configured to monitor access of data stored in a multi-class file system including a hierarchy of storage classes to generate access information for the data. For example, multiple classes of storage may be defined and data may be migrated automatically and transparently between the storage classes. In some embodiments, storage devices may be classified into different classes of storage to implement a multi-class file system. Each storage class includes one or more storage devices assigned to the storage class according to characteristics of the storage class. For example, storage classes may form a hierarchy, with the most expensive storage devices at the type and the least expensive at the bottom. A storage class may be based on cost, performance, business-value, or other characteristics of storage devices. According to some embodiments, a user may define multiple storage classes according to virtually any characteristics. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 3 – 30; page 8, line 16 – page 9, line 5; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 15, line 22 – page 16, line 26; page 22, line 10 – page 23, line 20; page 24, line 26 – page 25, line 30page 27, lines 18 – 24).

The multi-class storage mechanism is also configured to apply the access information to a set of policies for the multi-class file system. For instance, the multi-class storage mechanism may apply access information to a set of user-defined policies for initially assigning and migrating files in the hierarchy of storage classes, according to one embodiment. The multi-class storage mechanism is also configured to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to the application of the access information to the set of policies for the multi-class file system. For example, in one embodiment, less frequently accessed files maybe migrated to a storage class including lower performing storage devices. Thus, in response to applying access information (e.g. how frequently a file is accessed) to the set

of polices, the multi-class storage mechanism may migrate data to different storage classes in the hierarchy. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 5, lines 2 – 15; page 8, lines 3 – 14; page 9, lines 7 – 23; page 11, line 28 – page 12, line 2; page 27, lines 18 – 24; page 12, lines 8 – 16; page 13, lines 2 – 9).

Additionally, the migrated data remains online within the multi-class file system. For example, the data may remain online as part of the file system and the migration of the data may be transparent to a user or application. In other words, migrated files may still be stored in the same logical storage location within a file system. For instance, in some embodiments, file system metadata may be used to track the storage and migration of data in a multi-class file system. When a file (or portion of a file) is migrated to a storage class, the file system metadata may be modified to indicate the location of the data. For the perspective of an application or user, the path to the file may not change when the metadata is modified to reflect the migration of the data, according to some embodiments. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 4, lines 4 – 10 and lines 25 – 30; page 5, lines 12 – 16 and lines 24 - 29; page 8, lines 2 – 10; page 9, lines 3 – 23; page 12, lines 27 – 30; page 13, line 21 – page 14, line 2; page 14, lines 19 – 26; page 15, lines 3 – 8; page 17, lines 8 – 22; page 18, line 22 – page 19, lines 6; page 21, lines 18 – 28; page 23, lines 2 – 20; page 27, line 26 – page 28, line 9).

Independent claim 14 is directed to a system including a plurality of storage devices and a host system configured to couple to the storage devices via a network. The host system includes file system software comprising file system functionality and a multi-class storage mechanism. (*See, e.g.*, items 102, 104 and 106 of FIGs. 1 and 2; items 142, 144 and 146 of FIG. 3; items 202, 204, and 206 of FIG. 4; items 250, 252, 254, 256, 260 and 258 of FIG 5; page 8, line 16 – page 9, line 5; page 15, line 22 – page 16, line 26; page 22, line 10 – page 23, line 20; page 24, line 26 – page 25, line 30).

The file system functionality is configured to implement a multi-class file system including a hierarchy of storage classes on the plurality of storage devices. For example, multiple classes of storage may be defined and data may be migrated automatically and

transparently between the storage classes. In some embodiments, storage devices may be classified into different classes of storage to implement a multi-class file system. Each storage class includes one or more of the storage devices assigned to the storage class according to characteristics of the storage class. For example, storage classes may form a hierarchy, with the most expensive storage devices at the type and the least expensive at the bottom. A storage class may be based on cost, performance, business-value, or other characteristics of storage devices. According to some embodiments, a user may define multiple storage classes according to virtually any characteristics. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 3 – 30; page 8, line 25 – page 9, line 5; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 27, lines 18 – 24).

The multi-class storage mechanism is configured to monitor access of data stored in the multi-class file system to generate access information for the data and apply the access information to a set of policies for the multi-class file system. For example, a multi-class storage mechanism may monitor file usage, read and/or write access frequency, and other access characteristics and may migrate files (or portions of files) according to the application of the access information to a set of (possibly user defined) policies, according to one embodiment. The multi-class storage mechanism is also configured to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to said application of the access information to a set of policies for the multi-class file system. For instance, in one embodiment, data that have not been written to for some period may be migrated to a lower class of storage according to the policies. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 4 – 10 and lines 25 – 30; page 5, lines 12 – 16 and lines 24 - 29; page 8, lines 2 – 10; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 23, lines 2 – 20; page 24, lines 7 – 18; page 27, lines 18 – 24; page 29, lines 10 – 25; page 31, lines 1 – 19).

Additionally, the migrated data remains online within the multi-class file system. As described above, the data may remain online as part of the file system and the migration of the data may be transparent to a user or application. In other words, migrated files may still be stored in the same logical storage location within a file system.

For instance, in some embodiments, file system metadata may be used to track the storage and migration of data in a multi-class file system. When a file (or portion of a file) is migrated to a storage class, the file system metadata may be modified to indicate the location of the data. For the perspective of an application or user, the path to the file may not change when the metadata is modified to reflect the migration of the data, according to some embodiments. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 4, lines 4 – 10 and lines 25 – 30; page 5, lines 12 – 16 and lines 24 - 29; page 8, lines 2 – 10; page 9, lines 3 – 23; page 12, lines 27 – 30; page 13, line 21 – page 14, line 2; page 14, lines 19 – 26; page 15, lines 3 – 8; page 17, lines 8 – 22; page 18, line 22 – page 19, lines 6; page 21, lines 18 – 28; page 23, lines 2 – 20; page 27, line 26 – page 28, line 9).

Independent claim 15 is directed to a system including means for implementing a multi-class file system comprising a hierarchy of storage classes on a plurality of storage devices. Each storage class comprises one or more of the storage devices assigned to the storage class according to characteristics of the storage class. The structure corresponding to the means for implementing the multi-class file system comprising a hierarchy of storage classes on a plurality of storage devices is, for example, a processor, such as processor 250, coupled to a memory comprising program instructions executable by the processor, such as memory 254, illustrated in FIG. 5. Memory 254 may be representative of various types of possible memory media, such as flash memory, RAM, DRAM, SRAM EDO RAM, SDRAM, DDR SDRAM, etc. (*See, e.g.*, items 102, 104 and 106 of FIGs. 1 and 2; items 142, 144 and 146 of FIG. 3; items 202, 204, and 206 of FIG. 4; items 250, 252, 254, 256, 260 and 258 of FIG 5; page 8, line 16 – page 9, line 5; page 15, line 22 – page 16, line 26; page 22, line 10 – page 23, line 20; page 24, line 26 – page 25, line 30).

For example, multiple classes of storage may be defined and data may be migrated automatically and transparently between the storage classes. In some embodiments, storage devices may be classified into different classes of storage to implement a multi-class file system. Each storage class includes one or more storage devices assigned to the storage class according to characteristics of the storage class. For

example, storage classes may form a hierarchy, with the most expensive storage devices at the type and the least expensive at the bottom. A storage class may be based on cost, performance, business-value, or other characteristics of storage devices. According to some embodiments, a user may define multiple storage classes according to virtually any characteristics. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 3 – 30; page 8, line 25 – page 9, line 5; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 27, lines 18 – 24).

The system of claim 15 also includes software means for assigning and migrating data to different storage classes in the hierarchy of storage classes according to a set of policies for the multi-class file system. The structure corresponding to the means for assigning and migrating data to different storage classes in the hierarchy of storage classes according to a set of policies for the multi-class file system is, for example, a processor and memory comprising program instructions, such as processor 250 and memory 254 illustrated in FIG. 5. (*See, e.g.*, items 102, 104 and 106 of FIGs. 1 and 2; items 142, 144 and 146 of FIG. 3; items 202, 204, and 206 of FIG. 4; items 250, 252, 254, 256, 260 and 258 of FIG 5; page 8, line 16 – page 9, line 5; page 15, line 22 – page 16, line 26; page 22, line 10 – page 23, line 20; page 24, line 26 – page 25, line 30).

For instance, the multi-class storage mechanism may apply access information to a set of user-defined policies for initially assigning and migrating files in the hierarchy of storage classes, according to one embodiment. The multi-class storage mechanism is also configured to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to the application of the access information to the set of policies for the multi-class file system. For example, in one embodiment, less frequently accessed files maybe migrated to a storage class including lower performing storage devices. Thus, in response to applying access information (e.g. how frequently a file is accessed) to the set of polices, the multi-class storage mechanism may migrate data to different storage classes in the hierarchy. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 5, lines 2 – 15; page 8, lines 3 – 14; page 9, lines 7 – 23; page 11, line 28 – page 12, line 2; page 27, lines 18 – 24; page 12, lines 8 – 16; page 13, lines 2 – 9).

Independent claim 16 is directed to a method including multi-class storage mechanism software monitoring access of data stored in a multi-class file system comprising a hierarchy of storage classes to generate access information for the data. Each storage class includes one or more storage devices assigned to the storage class according to one or more characteristics of the storage class. For example, multiple classes of storage may be defined and data may be migrated automatically and transparently between the storage classes. In some embodiments, storage devices may be classified into different classes of storage to implement a multi-class file system. Each storage class includes one or more storage devices assigned to the storage class according to characteristics of the storage class. For example, storage classes may form a hierarchy, with the most expensive storage devices at the type and the least expensive at the bottom. A storage class may be based on cost, performance, business-value, or other characteristics of storage devices. According to some embodiments, a user may define multiple storage classes according to virtually any characteristics. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 3 – 30; page 8, line 25 – page 9, line 5; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 27, lines 18 – 24).

The method of claim 16 also includes the multi-class storage mechanism software applying the access information to a set of policies for the multi-class file system. For instance, the multi-class storage mechanism may apply access information to a set of user-defined policies for initially assigning and migrating files in the hierarchy of storage classes, according to one embodiment. The multi-class storage mechanism is also configured to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to the application of the access information to the set of policies for the multi-class file system. For example, in one embodiment, less frequently accessed files maybe migrated to a storage class including lower performing storage devices. Thus, in response to applying access information (e.g. how frequently a file is accessed) to the set of polices, the multi-class storage mechanism may migrate data to different storage classes in the hierarchy. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 5, lines 2 – 15; page 8, lines 3 – 14; page 9, lines 7 – 23; page 11, line 28 – page 12, line 2; page 27,

lines 18 – 24; page 12, lines 8 – 16; page 13, lines 2 – 9).

The method of claim 16 also includes migrating a portion of the data to different storage classes in the hierarchy of storage classes in response to applying the accessing information to the set of policies for the multi-class file system. For example, in one embodiment, less frequently accessed files maybe migrated to a storage class including lower performing storage devices. Thus, in response to applying access information (e.g. how frequently a file is accessed) to the set of polices, the multi-class storage mechanism may migrate data to different storage classes in the hierarchy. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 5, lines 2 – 15; page 8, lines 3 – 14; page 9, lines 7 – 23; page 11, line 28 – page 12, line 2; page 27, lines 18 – 24; page 12, lines 8 – 16; page 13, lines 2 – 9).

Additionally, the migrated data remains online within the multi-class file system. As described above, the data may remain online as part of the file system and the migration of the data may be transparent to a user or application. In other words, migrated files may still be stored in the same logical storage location within a file system. For instance, in some embodiments, file system metadata may be used to track the storage and migration of data in a multi-class file system. When a file (or portion of a file) is migrated to a storage class, the file system metadata may be modified to indicate the location of the data. For the perspective of an application or user, the path to the file may not change when the metadata is modified to reflect the migration of the data, according to some embodiments. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 4, lines 4 – 10 and lines 25 – 30; page 5, lines 12 – 16 and lines 24 - 29; page 8, lines 2 – 10; page 9, lines 3 – 23; page 12, lines 27 – 30; page 13, line 21 – page 14, line 2; page 14, lines 19 – 26; page 15, lines 3 – 8; page 17, lines 8 – 22; page 18, line 22 – page 19, lines 6; page 21, lines 18 – 28; page 23, lines 2 – 20; page 27, line 26 – page 28, line 9).

Independent claim 30 is directed to a computer-accessible storage medium comprising program instructions that are configured to implement monitoring access of data stored in a multi-class file system including a hierarchy of storage classes to generate access information for the data. Each storage class includes one or more storage devices

assigned to the storage class according to one or more characteristics of the storage class. For example, multiple classes of storage may be defined and data may be migrated automatically and transparently between the storage classes. In some embodiments, storage devices may be classified into different classes of storage to implement a multi-class file system. Each storage class includes one or more storage devices assigned to the storage class according to characteristics of the storage class. For example, storage classes may form a hierarchy, with the most expensive storage devices at the type and the least expensive at the bottom. A storage class may be based on cost, performance, business-value, or other characteristics of storage devices. According to some embodiments, a user may define multiple storage classes according to virtually any characteristics. (*See, e.g.*, FIGs. 1, 2, 3, 5, 6, and 7A-E; page 4, lines 3 – 30; page 8, line 25 – page 9, line 5; page 10, lines 6 – 30; page 11, lines 2 – 13 and 19-28; page 27, lines 18 – 24).

The program instructions are also configured to implement applying the access information to a set of policies for the multi-class file system and migrating a portion of the data to different storage classes in the hierarchy of storage classes in response to applying the access information and other file information to the set of policies for the multi-class file system. For instance, the multi-class storage mechanism may apply access information to a set of user-defined policies for initially assigning and migrating files in the hierarchy of storage classes, according to one embodiment. The multi-class storage mechanism is also configured to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to the application of the access information to the set of policies for the multi-class file system. For example, in one embodiment, less frequently accessed files maybe migrated to a storage class including lower performing storage devices. Thus, in response to applying access information (e.g. how frequently a file is accessed) to the set of polices, the multi-class storage mechanism may migrate data to different storage classes in the hierarchy. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 5, lines 2 – 15; page 8, lines 3 – 14; page 9, lines 7 – 23; page 11, line 28 – page 12, line 2; page 27, lines 18 – 24; page 12, lines 8 – 16; page 13, lines 2 – 9).

Additionally, the migrated data remains online within the multi-class file system. As described above, the data may remain online as part of the file system and the migration of the data may be transparent to a user or application. In other words, migrated files may still be stored in the same logical storage location within a file system. For instance, in some embodiments, file system metadata may be used to track the storage and migration of data in a multi-class file system. When a file (or portion of a file) is migrated to a storage class, the file system metadata may be modified to indicate the location of the data. For the perspective of an application or user, the path to the file may not change when the metadata is modified to reflect the migration of the data, according to some embodiments. (*See, e.g.*, FIGs. 2, 3, 6, 7A-E; page 4, lines 4 – 10 and lines 25 – 30; page 5, lines 12 – 16 and lines 24 - 29; page 8, lines 2 – 10; page 9, lines 3 – 23; page 12, lines 27 – 30; page 13, line 21 – page 14, line 2; page 14, lines 19 – 26; page 15, lines 3 – 8; page 17, lines 8 – 22; page 18, line 22 – page 19, lines 6; page 21, lines 18 – 28; page 23, lines 2 – 20; page 27, line 26 – page 28, line 9).

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 2, 6-12, 14-18, 22-28, 30 and 34-40 stand finally under 35 U.S.C. § 102(e) as being anticipated by Yakir et al. (U.S. Publication 2004/0049513) (hereinafter "Yakir").

2. Claims 3-5, 13, 19-21, 29, 31-33 and 41 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Yakir in view of Leung et al. (U.S. Publication 2004/0039891) (hereinafter "Leung").

3. Claims 5, 21 and 33 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Yakir in view of Gill (U.S. Patent 6,947,959).

VII. <u>ARGUMENT</u>

<u>First Ground of Rejection</u>

Claims 1, 2, 6-12, 14-18, 22-28, 30 and 34-40 stand finally under 35 U.S.C. § 102(e) as being anticipated by Yakir et al. (U.S. Publication 2004/0049513) (hereinafter "Yakir"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

<u>Claims 1, 8 - 10, 12, 14, 16, 24 - 26, 28, 30, 36 – 38 and 41</u>:

Regarding claim 1, Yakir does not disclose <u>file system</u> software comprising a multi-class storage mechanism, wherein the multi-class storage mechanism is configured to monitor access of data <u>stored in a multi-class file system</u> comprising <u>**a hierarchy of storage classes**</u> to generate access information for the data, wherein <u>each storage class comprises one or more storage devices assigned to the storage class</u> <u>**according to one or more characteristics of the storage class**</u>. Yakir teaches a multi-disk and multi-volume system, but does not disclose a hierarchy of storage classes where each storage class comprises storage devices assigned to the storage class *according to characteristics of the storage class*. The Examiner cites paragraphs [0020], [0070], [0090] and [0092] of Yakir, asserting that Yakir's "storage units 102 may be organized into one or more logical storage units/devices 104" and that a "logical storage unit may reside on non-continuous physical partitions." However, Yakir does not mention anything about a multi-class file system including *a hierarchy of storage classes*, where each storage class includes storage devices assigned to the storage class *according to characteristics of the storage class*. Instead, Yakir merely discloses multiple storage devices and multiple logical storage units. Traditionally, multiple storage devices and logical storage unit have been used to store data without using a hierarchy of storage classes and without having each storage class include storage devices assigned to the storage class according to characteristics of the storage class. The fact that Yakir's system includes multiple storage devices/units does not disclose the specific limitations of a multi-class file system

including *a hierarchy of storage classes* and storage devices assigned to a storage class *according to characteristics of the storage class*.

In the Response to Argument, the Examiner argues that Yakir's servers (S1, S2 and S3) are equivalent to the storage classes of Applicants' claims. The Examiner is incorrect. Yakir does not teach that the servers of his system represent a <u>hierarchy of storage classes in which each storage class includes storage devices assigned to the storage class, according to characteristics of the storage class</u>. Additionally, the Examiner equates Yakir's logical storage units with the storage devices of Applicants' claims. However, Yakir specifically teaches that a "single logical storage unit may span storage space provided by multiple physical storage units" and that a "single physical storage unit may be divided into several separately identifiable logical storage units" (paragraph [0020]). Yakir also clearly states that a physical storage unit, as opposed to a logical storage unit, "is intended to refer to any physical device, system, etc. that is capable of storing information or data" (paragraph [0019]). Thus, Yakir teaches that it's <u>logical</u> storage units are not storage <u>devices</u>, contrary to the Examiner's assertion.

The Examiner also refers to Yakir's mention of Hierarchical Storage Management (HSM) applications, citing paragraphs [0004] and [0046]. However, Applicants' argument is not that Yakir never mentions HSM applications, but that Yakir does not disclose a hierarchy of storage classes where *each storage class comprises storage devices* assigned to the storage class *according to characteristics of the storage class*, as argued above. Paragraphs [0004] and [0046] describe stub files, which Yakir describes as a physical file that represents a migrated file. Neither of the cited paragraphs ([0004] and [0046]) supports the Examiner contention that Yakir discloses a hierarchy of storage classes where *each storage class comprises storage devices* assigned to the storage class *according to characteristics of the storage class*. The fact that Yakir's system includes a stub file that "stores information that enables a migrated file to be recalled" does not in any way imply the use of a hierarchy of storage classes where *each storage class comprises storage devices* assigned to the storage class *according to characteristics of the storage class*, as recited in Applicants' claim.

The Examiner also asserts that each of Yakir's storage units "is generally identifiable by a unique identifier that may be specified by the administrator." However, providing a unique identifier for storage unit does not disclose assigning storage units to storage classes *according to one or more characteristics* *<u>of the storage class</u>*. The unique identifiers to which the Examiner refers merely allow each storage unit to be uniquely addressed. Yakir does not mention anything about the unique identifiers being characteristics of any storage class. Thus, Yakir fails to disclose wherein each storage class comprises one or more storage devices assigned to the storage class according to one or more characteristics of the storage class.

Furthermore, the Examiner argues in the Response to Arguments that Yakir's use of a unique identifier for each logical storage unit "discloses characteristics of the storage class", citing paragraph [0020]. The Examiner's has incorrectly interpreted the teachings of Yakir. Nowhere does Yakir describe a logical storage unit's identifier as representing any sort of characteristic of a storage class. The mere fact that a logical storage unit may be assigned a unique identifier by an administrator does not imply any sort of characteristic of the logical storage unit or of a storage class. Additionally, the Examiner's has already argued that Yakir's servers (S1, S2, and S3) are equivalent to storage classes and also argued (erroneously) that Yakir's logical storage units are equivalent to storage devices. Thus, the Examiner is contradicting herself. On one hand the Examiner argues that Yakir's logical storage units are storage devices (and that the servers are storage classes) and on the other hand the Examiner argues that a logical storage unit's unique identifier is a characteristic of a storage class. The Examiner cannot have it both ways. The unique identifier for a logical storage unit, which the Examiner considers a storage device, has nothing to do with Yakir's servers and thus cannot also be a characteristic of a storage class, which the Examiner considers to be equivalent to Yakir's servers.

Further regarding claim 1, **Yakir also fails to disclose a multi-class storage mechanism configured to <u>apply the access information to a set of policies for the</u>**

multi-class file system. The Examiner cites FIG. 1, item 114 and paragraph [0023] of Yakir. However, item 114 of FIG. 1 and paragraph [0023] merely disclose that Yakir's system includes "information 114 related to storage policies and rules configured for the storage environment" (Yakir, paragraph [0023]). Yakir does not, however, teach anything regarding applying access information (generated from the monitoring of data stored in a multi-class file system) to a set of policies for the multi-class file system. Yakir does not teach anything regarding applying any access information to the storage policies and rules of information 114. Nor does Yakir describe applying access information to any other set of policies. The mere existence of storage policies does not inherently include or imply applying access information to storage policies. Without some specific disclosure by Yakir regarding applying access information to a set of policies, Yakir cannot be said to anticipate a multi-class storage mechanism configured to apply access information to a set of policies for a multi-class file system.

In the response to arguments, the Examiner cites paragraphs [0023] and [0046] and argues that in Yakir's system information used to find or locate migrated data may be stored in the same database as "information related to storage policies and rules configured for the storage environment". However, the sentence cited by the Examiner is the only reference by Yakir to such polices or rules. Thus, the Examiner is arguing that since Yakir mentions that information used to locate migrated data may be stored together with (in the same database as) "information related to storage policies", Yakir somehow *discloses* the specific limitation of *applying access information to a set of policies* for a multi-class file system, as recited in Applicants' claim. **The Examiner's position clearly goes beyond the actual teachings of Yakir.** A single sentence stating that location information and "information related to policies" may be stored in the same database does not, in any way, *disclose* applying access information to a set of policies. Storing different types of information together does not imply that one set of information is *applied* to the other.

Additionally, Yakir fails to disclose a multi-class storage mechanism configured to migrate a portion of the data to different storage classes in the

hierarchy of storage classes <u>in response to the application of access information to the set of policies</u> for the multi-class file system. The Examiner cites the same portions of Yakir (FIG. 1, item 114 and paragraphs [0020], [0023], [0070], [0090] and [0092]). However, none of the cited passages mentions anything about migrating data to different storage classes *in response to the application of access information* to the set of policies. The Examiner refers to Yakir's teachings regarding migrating a stub file from one storage unit to another, but fails to cite any portion of Yakir that discloses migrating a stub file **in response to the application of access information to a set of policies**. Instead, Yakir teaches that a stub file is migrated in response to an originating server receiving a signal to move a stub file and that "[t]he signal may be received from a user, an application or program, or from other like source" (Yakir, paragraph [0063]). Thus, Yakir discloses migrating a stub file in response to a signal from a user, application or a similar source. A signal from a user or an application cannot be considered an application of access information to a set of policies. Yakir clearly does not describe migrating data in response to the application of access information to a set of policies.

In the Response to Arguments, the Examiner cites paragraph [0011] of Yakir referring to Yakir's teachings that "information (such as information 11 related to policies) can be used to determine the location of the migrated data" (parenthesis by Examiner). However, the Examiner's argument fails to support the Examiner's position. Firstly, the cited paragraph does not support the Examiner's contention that "information related to polices" can be used to determine the location of migrated data. Instead, paragraph [0011] states that a stub file stores information that can be used to determine the location of migrated data. Secondly, the fact that information (whether related to policies or not) may be used to locate *migrated* data, i.e. data that has already be migrated does not disclose migrating data to different storage classes *in response to* the application of access information to a set of policies. The Examiner's argument that after being migrated, information related to policies may be used to locate the migrated data says nothing about whether the data was migrated in response to anything.

Anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Yakir fails to disclose a multi-class storage mechanism configured to monitor access of data stored in a multi-class file system comprising a hierarchy of storage classes to generate access information for the data, wherein each storage class comprises one or more storage devices assigned to the storage class according to one or more characteristics of the storage class. Yakir further fails to disclose that the multi-class storage mechanism is configured to apply the access information to a set of policies for the multi-class file system and to migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to the application of access information to the set of policies for the multi-class file system. Therefore, Yakir cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claims 14, 16 and 30.

Claims 2 and 18:

Regarding claim 2, Yakir fails to disclose file system software that includes file system functionality configured to implement the hierarchy of storage classes of the multi-class file system. The Examiner cites paragraphs [0020], [0070], [0090] and [0092] of Yakir and asserts that in Yakir's system, "[p]hysical storage units 102 may be organized into one or more logical storage units/devices 104" and that a "logical storage unit may reside on non-[contiguous] physical partitions." However, as noted above regarding the rejection of claim 1, merely having multiple physical and logical storage units where a logical storage unit may reside on non-contiguous physical partitions does not disclose or imply a hierarchy of storage classes. Thus, Yakir fails to disclose a

hierarchy of storage classes. Additionally, Yakir fails to disclose file system functionality configured to implement the hierarchy of storage classes. A file system is a particular type of software that performs specific function, as is understood in the art. Yakir only teaches that data may be migrated from an original storage location to another storage location but makes no mention of any file system functionality configured to implement a hierarchy of storage classes. Thus, the rejection of claim 2 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 18.

Claims 6, 22 and 34:

In Regards to claim 6, **Yakir fails to disclose where the multi-class storage mechanism is configured to initially place the data in the storage classes in the hierarchy of storage classes according to the set of policies**. The Examiner cites FIG. 1 and paragraphs [0020], [0023], [0070], [0090] and [0092] of Yakir, referring to the fact that Yakir's system includes storage policy information 114 that may include storage environment. The only thing that Yakir mentions about storage policy information 114 is that database 112 may "include information 114 related to storage policies and rules configured for the storage environment" (Yakir, paragraph [0023]).

Yakir never describes initially placing data into storage classes in a hierarchy of storage classes according to storage policy information 114 or any other set of polices. As described above regarding the rejection of claim 1, Yakir fails to mention anything regarding a hierarchy storage classes. Yakir also fails to describe storage policy information 114 as including information regarding placing data within a hierarchy of storage classes. Yakir does not disclose anything regarding how or when the storage polices and rules in storage policy information 114 might be used. Without some specific disclosure regarding placing information in a hierarchy of storage classes according to storage policy information 114, the Examiner is merely speculating (improperly, in hindsight) regarding how Yakir's system might work.

Thus, the Examiner rejects claim 6 relying solely on the fact that Yakir mentions "storage policies and rules configured for the storage environment", without considering the specific language and limitations of claim 6. For at least the reasons presented above, the rejection of claim 6 is not supported by the cited art and removal thereof is respectfully requested.

Claims 7, 23 and 35:

Regarding claim 7, Yakir fails to disclose where the multi-class storage mechanism is configured to modify file system metadata for the migrated data to indicate the different storage classes for the migrated data. The Examiner cites paragraphs [0049-0053] of Yakir. This portion of Yakir describes that data, metadata and stub files may be migrated but does not mention anything regarding modifying metadata to indicate different storage classes for migrated data. Yakir does not disclose any kind of indication of different storage classes for migrated data, either in metadata or elsewhere. Merely describing that data, metadata and stub files can be migrated does not disclose the specific limitation of modifying file system metadata to indicate different storage classes for the migrated data.

In the Response to Arguments, the Examiner cites paragraphs [0067] and [0078] of Yakir, referring to Yakir teaching regarding "the originating server modify/update information stored in a database". However, the sentence cited by the Examiner (in both paragraph [0067] and paragraph [0078]) actually states, "[t]he originating server may modify/update information stored in a database ... using database update techniques such as ODBC techniques." In fact, both cited paragraphs are about techniques used to inform storage management server (SMS) 110 regarding a stub file. In one case (paragraph [0067]) Yakir teaches that SMS 110 is informed that a stub file now resides on a different volume and in the other case (paragraph [0078]) Yakir teaches that SMS 110 is informed that a stub file is no longer eligible for remigration. Nothing in the cited passages has any anything to do with modified file system metadata for migrated data to indicate the different storage classes for the migrated data. Informing a server that a stub file is on a

different volume does not imply that the different volume is on a different storage class and information a server that a stub file is ineligible for remigration also fails to imply anything about indicating different storage classes for the migrated data.

Thus, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 23 and 35.

<u>Claims 11, 27 and 39</u>:

Regarding claim 11, Yakir fails to disclose file system software that is configured to <u>add a new storage class to the hierarchy of storage classes</u>. The Examiner cites paragraph [0006] of the background section of Yakir and refers to Yakir's teaching regarding reorganizing data when deploying new servers. However, the cited passage of Yakir does not describe anything about adding a new storage class to a hierarchy of storage classes. Instead, the cited passage describes how stub files may be moved from one storage location to another for various reasons, including reorganizing data when deploying new servers and storage devices. No mention is made about *adding a new storage class* to a hierarchy of storage classes. In fact, nowhere does Yakir mention file system software configured to add a new storage class to the hierarchy of storage classes.

In the Response to Arguments the Examiner argues that Yakir teaches "a new destination storage location", citing paragraphs [0006 – 0007] of Yakir. Applicants respectfully disagree with the Examiner's interpretation. The cited passage describes that when a "user moves a stub file from its present location to a new destination storage location" conventional HSM application "always recall the file data corresponding to the stub file from the repository storage location …". The cited passage has nothing to do with adding a new storage class to the hierarchy of storage classes.

Merely describing that a stub file may be moved "from its present location to a new destination storage location" does not disclose file system software that is configured to <u>add a new storage class to the hierarchy of storage classes</u>. Moving a stub file, or other

data, to a new destination does not in any way imply that a new storage class is added to a hierarchy of storage classes. Data can be moved between locations within a single volume, devices, and certainly with a single storage class. Moreover, even if data is moved from one storage class to another, that does not necessarily involve adding a new storage class to a hierarchy of storage classes. The fact that Yakir teaches that a stub file may be moved to a "new destination storage location" does not disclose file system software that is configured to <u>add a new storage class to the hierarchy of storage classes</u>.

Thus, for at least the reasons above, the rejection of claim 11 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 27 and 39.

<u>Claim 15</u>:

Regarding claim 15, Yakir fails to disclose a system including means for implementing a multi-class file system including <u>a hierarchy of storage classes</u> on a plurality of storage devices, where each storage class includes one or more of the storage devices <u>assigned to the storage class according to one or more **characteristics of the storage class**</u>. Please refer to the remarks above regarding claim 1, for a detailed discussion of Yakir's failure to disclose a multi-class file system including <u>a hierarchy of storage classes</u> on a plurality of storage devices, where each storage class includes one or more of the storage devices <u>assigned to the storage class according to one or more **characteristics of the storage class**</u>.

Yakir further fails to disclose software means for assigning migrating data <u>to different storage classes in the hierarchy of storage classes **according to a set of policies**</u> for the multi-class file system. The Examiner does not cite any portion of Yakir that describes migrating data to different storage classes in a hierarchy of storage classes. Yakir only mentions that data may be migrated from an original storage location on an original volume to a repository storage location on a repository volume and that a stub file may also be migrated from an original storage location to another storage location.

However, Yakir does not mention migrating data to *different storage classes in a hierarchy of storage classes*. In fact, Yakir makes no mention of different storage classes at all. The Examiner equates the mere fact that Yakir's system includes multiple physical storage devices and multiple logical storage units as including a hierarchy of storage classes. However, merely providing multiple physical storage devices and multiple logical storage units does not disclose anything regarding *different storage classes* or about a *hierarchy* of storage classes.

Nor does having multiple physical and logical storage units disclose anything about assigning and migrating data according to a set of policies for a multi-class file system. Yakir merely describes the existence of storage policies and rules (Yakir, paragraph [0023]), but fails to disclose migrating data to different storage classes according to a set of policies. As noted above regarding claim 1, Yakir describes migrating a stub file in response to receiving a signal from a user, application, program, or other like sources. Nowhere does Yakir mention anything regarding migrating data according to a set of policies.

Thus, the rejection of claim 15 is not supported by the cited art and removal thereof is respectfully requested.

Second Ground of Rejection

Claims 3-5, 13, 19-21, 29, 31-33 and 41 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Yakir in view of Leung et al. (U.S. Publication 2004/0039891) (hereinafter "Leung"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 3, 19 and 31:

Regarding claim 3, Yakir in view of Leung does not teach or suggest **storage classes that are ordered in a hierarchy of storage classes according to performance characteristics from a highest storage class comprising high-performance storage devices to a lowest storage class comprising low-performance storage devices**. The Examiner admits that Yakir does not teach storage classes ordered in a hierarchy according to performance characteristics but relies upon Leung, citing paragraphs [0037-0038] and [0053-0054]. However, none of the cited paragraphs mentions storage classes ordered in a hierarchy of storage classes *according to performance characteristics*. Instead, the cited paragraphs describe storage units may be classified into groups according to the data storage cost, such as a monetary value of storage data per unit of storage. Yakir also describes using other criteria, such as volume capacity, manufacturer, or device type, to group storage units. However, Leung, whether considered singly or in combination with Yakir, fails to teach or suggest anything regarding storage classes ordered *according to performance characteristics*.

In the Response to Arguments, the Examiner cites paragraphs [0070] and [0091] and arguing that "rank" and "order" are equivalent. First of all the Examiner fails to state whether the cited paragraphs are from Yakir or Leung. Furthermore, paragraphs [0070] and [0091] of both Yakir and Leung fail to mention anything regarding storage classes that are ordered (or ranked) in a hierarchy according to performance characteristics from a highest storage class comprising high-performance storage devices to a lowest storage class comprising low-performance storage devices.

Paragraph [0070] of Yakir states that a stub file may be moved "without recalling migrated data associated with the stub file". Paragraph [0091] of Yakir describes that data locator information is stored in a stub file and that moving the stub file from the originating storage location to the destination storage location essentially moves the information stored by the stub file from the originating storage location to the destination storage location.

Paragraph [0070] of Leung states that when an overcapacity condition (e.g., when the used storage capacity for a volume exceeds a user-configured threshold value) is detected, a target volume is then automatically and dynamically determined for receiving files from the source volume to resolve the overcapacity condition and that data is moved from a source volume to a target volume that has a lower storage data cost associated with it. However, moving data to a volume that has lower storage data cost does not teach or suggest storage classes that are ordered in a hierarchy of storage classes according to performance characteristics from a highest storage class comprising high-performance storage devices to a lowest storage class comprising low-performance storage devices. Storage cost and performance are two different things.

Paragraph [0091] of Leung describes file groups. Leung teaches that a file is included in a file group if the file satisfies criteria specified for the file group and that the file group criteria may be specified by the administrator or some other user. Leung gives the example that an administrator may create file groups based upon a business value associated with the files where the administrator may group files that are deemed important or critical for the business into one file group (a "more important" file group) and the other files may be grouped into a second group (a "less important" file group). However, grouping files is not the same as storage classes that are ordered in a hierarchy of storage classes according to performance characteristics. Additionally, the Examiner has previously argued that Yakir's servers S1, S2 and S3, not Leung file groups are storage classes. Grouping files as taught by Leung does not teach or suggest ordering storage classes (or Yakir's servers) according to performance characteristics.

Yakir and Leung, whether considered singly or in combination, fail to teach or suggest storage classes that are ordered in a hierarchy of storage classes according to performance characteristics from a highest storage class comprising high-performance storage devices to a lowest storage class comprising low-performance storage devices. Thus, for at least the reasons above, the rejection of claim 3 is not supported by the cited

art and removal thereof is respectfully requested. Similar remarks also apply to claims 19 and 31.

Claims 4, 20 and 32:

Regarding claim 4, Yakir in view of Leung fails to teach or suggest **wherein the multi-class storage mechanism is further configured to migrate less-frequently-accessed data to lower storage classes comprising lower-performing storage devices and to migrate more-frequently-accessed data to higher storage classes comprising higher-performing storage devices according to the set of policies**. The Examiner admits that Yakir fails to teach or suggest this limitation of claim 4 and relies on Leung, citing various paragraphs. However, none of the cited passages describes migrating less frequently accessed data to lower storage classes including lower-performing storage devices. Leung also fails to describe migrating more frequently accessed data to higher storage classes including higher-performing storage devices.

The Examiner refers to Leung's teaching regarding file groups. As noted above regarding the rejection of claim 3, Leung teaches that a file is included in a file group if the file satisfies criteria specified for the file group and that the file group criteria may be specified by the administrator or some other user. Leung gives the example that an administrator may create file groups based upon a business value associated with the files where the administrator may group files that are deemed important or critical for the business into one file group (a "more important" file group) and the other files may be grouped into a second group (a "less important" file group).

However, Leung does not teach or suggest migrating less-frequently-accessed data to storage classes including lower-performing storage devices and migrating more-frequently-accessed data to storage classes with higher-performing storage devices. Instead, Leung teaches moving data from higher cost storage to lower cost storage (Yakir, paragraphs [0012 - 0014], [0054], [0058], [0070]). Migrating data from higher cost storage to lower cost storage does not teach, suggest or imply migrating less-

frequently-accessed data to storage classes including lower-performing storage devices and migrating more-frequently-accessed data to storage classes with higher-performing storage devices.

Furthermore, the Examiner has failed to provide a proper motivation for combining Yakir and Leung. The Examiner states, "[t]he ordinarily skilled artisan would have been motivated to modify Yakir for the purpose of conveniently assisting the user by providing data usage information". Providing data usage information has nothing to with migrating less-frequently-accessed data to storage classes including lower-performing storage devices and migrating more-frequently-accessed data to storage classes with higher-performing storage devices. Thus, the Examiner's stated motivation would not motivate anyone to modify the migrating policies of Yakir to include the file groups of Leung.

Thus, Yakir and Leung, whether considered singly of in combination, fail to teach or suggest **wherein the multi-class storage mechanism is further configured to <u>migrate less-frequently-accessed data</u> <u>to lower storage classes comprising lower-performing storage devices</u> and to <u>migrate more-frequently-accessed data to higher storage classes comprising higher-performing storage devices</u> according to the set of policies**. For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested.

<u>Claims 5, 21 and 33</u>:

The Examiner has filed to provide a proper prima facie rejection of claims 5, 21 and 33, under 35 U.S.C. § 103(a) as being unpatentable over Yakir in view of Leung. The Examiner lists claims 5, 21 and 33 but fails to provide any actual arguments or citations regarding claims 5, 21 and 33 in the rejection over Yakir in view of Leung.

In regards to claim 5, **Yakir in view of Leung fails to teach or suggest compressing data migrated to one or more storage classes in the hierarchy of**

storage classes. Neither Yakir nor Leung, whether considered singly or in combination, mentions anything about compressing data migrated to storage classes in the hierarchy of storage classes. Furthermore, in the rejection of claim 5 over Yakir in view of Gill (see Third Ground of Rejection, below), the Examiner admits that Yakir does not teach or suggest compressing data migrated to storage classes in the hierarchy of storage classes. Leung fails to teach anything regarding compressing of migrated data. Thus, the Examiner's combination of Yakir and Leung also fails to teach or suggest compressing data migrated to one or more storage classes in the hierarchy of storage classes. For at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested.

Claims 13, 29 and 41:

Appellants traverse the rejection of claim 13, 29 and 41 for at least the reasons presented above regarding their respective, independent claims.

Third Ground of Rejection

Claims 5, 21 and 33 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Yakir in view of Gill (U.S. Patent 6,947,959). Appellants traverse this rejection for at least the reasons presented above regarding their respective, independent claims.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-41 was erroneous, and reversal thereof is respectfully requested.

No extension of time should be required since this Appeal Brief is filed within one month of the Notice of Panel Decision. The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-14900/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. # 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: November 13, 2006

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement file system software comprising a multi-class storage mechanism, wherein the multi-class storage mechanism is configured to:

monitor access of data stored in a multi-class file system comprising a hierarchy of storage classes to generate access information for the data, wherein each storage class comprises one or more storage devices assigned to the storage class according to one or more characteristics of the storage class;

apply the access information to a set of policies for the multi-class file system; and

migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to said application of the access information to the set of policies for the multi-class file system;

wherein the migrated data remains online within the multi-class file system.

2. The system as recited in claim 1, wherein the file system software further comprises File System functionality configured to implement the hierarchy of storage

classes of the multi-class file system.

3. The system as recited in claim 1, wherein the storage classes are ordered in the hierarchy of storage classes according to performance characteristics from a highest storage class comprising one or more high-performance storage devices to a lowest storage class comprising one or more low-performance storage devices.

4. The system as recited in claim 1, wherein the multi-class storage mechanism is further configured to migrate less-frequently-accessed data to lower storage classes comprising lower-performing storage devices and to migrate more-frequently-accessed data to higher storage classes comprising higher-performing storage devices according to the set of policies.

5. The system as recited in claim 1, wherein the multi-class storage mechanism is further configured to compress data migrated to one or more storage classes in the hierarchy of storage classes.

6. The system as recited in claim 1, wherein the multi-class storage mechanism is further configured to initially place the data in the storage classes in the hierarchy of storage classes according to the set of policies.

7. The system as recited in claim 1, wherein the multi-class storage mechanism is further configured to modify file system metadata for the migrated data to indicate the different storage classes for the migrated data, wherein path information in the file system metadata exposed to applications is not modified.

8. The system as recited in claim 1, wherein said migration of a portion of the data to the different storage classes is transparent to an application configured to access the multi-class file system.

9. The system as recited in claim 1, wherein the migrated data includes files

or portions of files.

10. The system as recited in claim 1, wherein the migrated data comprises one or more of application data and file system metadata.

11. The system as recited in claim 1, wherein the file system software is configured to add a new storage class to the hierarchy of storage classes, and wherein the multi-class storage mechanism is further configured to migrate data stored on one or more others of the storage classes to the new storage class according to the set of policies.

12. The system as recited in claim 1, wherein the file system software is configured to add a new storage device to a storage class in the hierarchy of storage classes, and wherein the multi-class storage mechanism is further configured to migrate data stored on one or more of the storage classes to the new storage device according to the set of policies.

13. The system as recited in claim 1, wherein the file system software is configured to increase the capacity allocated to a storage class on a storage device within the storage class.

14. A system, comprising:

a plurality of storage devices;

a host system configured to couple to the plurality of storage devices via a network, wherein the host system comprises file system software comprising:

File System functionality configured to implement a multi-class file system comprising a hierarchy of storage classes on the plurality of

storage devices, wherein each storage class comprises one or more of the storage devices assigned to the storage class according to one or more characteristics of the storage class; and

a multi-class storage mechanism configured to:

monitor access of data stored in the multi-class file system to generate access information for the data;

apply the access information to a set of policies for the multi-class file system; and

migrate a portion of the data to different storage classes in the hierarchy of storage classes in response to said application of the access information to a set of policies for the multi-class file system;

wherein the migrated data remains online within the multi-class file system.

15. A system, comprising:

means for implementing a multi-class file system comprising a hierarchy of storage classes on a plurality of storage devices, wherein each storage class comprises one or more of the storage devices assigned to the storage class according to one or more characteristics of the storage class;

software means for assigning and migrating data to different storage classes in the hierarchy of storage classes according to a set of policies for the multi-class file system.

16. A method, comprising:

multi-class storage mechanism software monitoring access of data stored in a multi-class file system comprising a hierarchy of storage classes to generate access information for the data, wherein each storage class comprises one or more storage devices assigned to the storage class according to one or more characteristics of the storage class;

the multi-class storage mechanism software applying the access information to a set of policies for the multi-class file system; and

migrating a portion of the data to different storage classes in the hierarchy of storage classes in response to said applying the access information to the set of policies for the multi-class file system;

wherein the migrated data remains online within the multi-class file system.

17. The method as recited in claim 16, wherein the multi-class storage mechanism software is part of file system software.

18. The method as recited in claim 16, further comprising File System functionality of file system software implementing the hierarchy of storage classes of the multi-class file system.

19. The method as recited in claim 16, wherein the storage classes are ordered in the hierarchy of storage classes according to performance characteristics from a highest storage class comprising one or more high-performance storage devices to a lowest storage class comprising one or more low-performance storage devices.

20. The method as recited in claim 16, wherein said migrating comprises:

migrating less-frequently-accessed data to lower storage classes comprising lower-performing storage devices; and

migrating more-frequently-accessed data to higher storage classes comprising higher-performing storage devices.

21. The method as recited in claim 16, further comprising compressing data migrated to one or more storage classes in the hierarchy of storage classes.

22. The method as recited in claim 16, further comprising the multi-class storage mechanism software initially placing the data in the hierarchy of storage classes according to the set of policies.

23. The method as recited in claim 16, further comprising the multi-class storage mechanism software modifying file system metadata to indicate the different storage classes for the migrated data, wherein path information in the file system metadata exposed to applications is not modified.

24. The method as recited in claim 16, wherein said migrating a portion of the data to different storage classes is transparent to an application that accesses the data in the hierarchy of storage classes.

25. The method as recited in claim 16, wherein the migrated data includes files or portions of files.

26. The method as recited in claim 16, wherein the migrated data comprises one or more of application data and file system metadata.

27. The method as recited in claim 16, further comprising:

adding a new storage class to the hierarchy of storage classes; and

the multi-class storage mechanism software transparently migrating data from one or more others of the storage classes to the new storage class.

28. The method as recited in claim 16, further comprising:

adding a new storage device to a storage class in the hierarchy of storage classes; and

the multi-class storage mechanism transparently migrating data stored on one or more of the storage classes to the new storage device.

29. The method as recited in claim 16, further comprising increasing the capacity allocated to a storage class on a storage device within the storage class.

30. A computer-accessible storage medium comprising program instructions, wherein the program instructions are configured to implement:

monitoring access of data stored in a multi-class file system comprising a hierarchy of storage classes to generate access information for the data, wherein each storage class comprises one or more storage devices assigned to the storage class according to one or more characteristics of the storage class;

applying the access information to a set of policies for the multi-class file system; and

migrating a portion of the data to different storage classes in the hierarchy of storage classes in response to said applying the access information and other file information to the set of policies for the multi-class file system;

wherein the migrated data remains online within the multi-class file system.

31. The computer-accessible storage medium as recited in claim 30, wherein the storage classes are ordered in the hierarchy of storage classes according to performance characteristics from a highest storage class comprising one or more high-performance storage devices to a lowest storage class comprising one or more low-performance storage devices.

32. The computer-accessible storage medium as recited in claim 30, wherein, in said migrating, the program instructions are further configured to implement:

migrating less-frequently-accessed data to lower storage classes comprising lower-performing storage devices; and

migrating more-frequently-accessed data to higher storage classes comprising higher-performing storage devices.

33. The computer-accessible storage medium as recited in claim 30, wherein the program instructions are further configured to implement compressing data migrated to one or more storage classes in the hierarchy of storage classes.

34. The computer-accessible storage medium as recited in claim 30, wherein the program instructions are further configured to implement initially placing the data in the hierarchy of storage classes according to the set of policies.

35. The computer-accessible storage medium as recited in claim 30, wherein the program instructions are further configured to implement modifying file system

metadata to indicate the different storage classes for the migrated data, wherein path information in the file system metadata exposed to applications is not modified.

36. The computer-accessible storage medium as recited in claim 30, wherein said migrating a portion of the data to different storage classes is transparent to an application that accesses the data in the hierarchy of storage classes.

37. The computer-accessible storage medium as recited in claim 30, wherein the migrated data includes files or portions of files.

38. The computer-accessible storage medium as recited in claim 30, wherein the migrated data comprises one or more of application data and file system metadata.

39. The computer-accessible storage medium as recited in claim 30, wherein the program instructions are further configured to implement:

adding a new storage class to the hierarchy of storage classes; and

transparently migrating data from one or more others of the storage classes to the new storage class.

40. The computer-accessible storage medium as recited in claim 30, wherein the program instructions are further configured to implement:

adding a new storage device to a storage class in the hierarchy of storage classes; and

the multi-class storage mechanism transparently migrating data stored on one or more of the storage classes to the new storage device.

41. The computer-accessible storage medium as recited in claim 30, wherein

the program instructions are further configured to implement increasing the capacity allocated to a storage class on a storage device within the storage class.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.